

Universitat Politècnica de Catalunya(UPC) - BarcelonaTech

Facultat d'Informàtica de Barcelona(FIB)

Grau en Enginyeria Informàtica
Especialitat d'Enginyeria del Software

LISP-DDT Sobre OpenOverlayRouter

Autor:

Ernest Puigdemont Velasco

Director:

Albert Cabellos Aparicio
(Arquitectura de Computadors)

Co-Director:

Jordi Paillissé Vilanova
(Arquitectura de Computadors)

24 de gener de 2018

Resum

Aquest projecte es basa en dissenyar i implementar la funcionalitat de LISP-DDT per a la plataforma OpenOverlayRouter(OOR). OOR és una implementació de codi obert per a crear xarxes de superposició programables, escrita en C i disponible per a Linux, Android i OpenWRT. Amb aquest projecte, incorporarà també LISP-DDT, completant així les funcionalitats de LISP.

Resumen

Este proyecto se basa en diseñar y implementar la funcionalidad de LISP-DDT para la plataforma OpenOverlayRouter(OOR). Oor es una implementación de código abierto para crear redes de superposición programables, escrita en C y disponible para Linux, Android y OpenWRT. Con este proyecto, incorporará también LISP-DDT, completando así las funcionalidades de LISP.

Abstract

This project consists in designing and implementing the functionalities of LISP-DDT for the OpenOverlayRouter(OOR) platform. OOR is an open source implementation to create programmable overlay networks, written in C and available for Linux, Android, and OpenWRT. This project will incorporate LISP-DDT to it, completing the LISP functionalities.

Índex

1 Introducció, abast i contextualització.....	5
1.1 Context i estat de l'art.....	5
1.1.1 Context.....	5
1.1.2 Actors implicats.....	8
1.1.3 Estat de l'Art.....	9
1.2 Objectius i abast.....	10
1.2.1 Objectius, formulació del problema.....	10
1.2.2 Abast.....	10
1.3 Metodologia i rigor.....	11
1.3.1 Metodologia.....	11
1.3.2 Validació.....	11
2 Planificació temporal.....	12
2.1 Duració del projecte.....	12
2.2 Diagrama Gantt.....	12
2.3 Tasques del projecte.....	13
2.4 Alternatives i pla d'acció.....	14
3 Gestió econòmica i sostenibilitat.....	15
3.1 Identificació i estimació dels costos.....	15
3.1.1 Recursos humans.....	15
3.1.2 Recursos hardware.....	15
3.1.3 Recursos software.....	15
3.1.4 Cost total.....	16
3.2 Control de gestió.....	16
3.3 Sostenibilitat i compromís social.....	16
Social.....	16
Ambiental.....	16
Taula.....	16
4 Desenvolupament del projecte.....	17
4.1 Decisions de disseny.....	17
4.2 Canvis respecte al RFC 8111.....	17
4.3 Esquema de classes.....	18
5 Resultats.....	19
5.1 Proves(llista).....	19
5.2 Proves(detall).....	21
5.2.1 Proves del DDT-NODE.....	21
5.2.2 Proves del DDT Map Server.....	25
5.2.3 Proves del DDT Map Resolver.....	31
6 Conclusions.....	38
Bibliografia.....	39

1 Introducció, abast i contextualització

1.1 Context i estat de l'art

1.1.1 Context

Què és LISP?

LISP(Locator/ID Separation Protocol)[1] és un protocol que especifica una arquitectura i mecanismes per a reemplaçar les adreces actualment utilitzades per IP amb dos espais de nom separats:

- Endpoint Identifiers(EIDs), que s'utilitzen per a finalitzar associacions de la capa de transport
- Routing Locators(RLOCs), que estan lligats a una localització, i s'utilitzen per a enrutar i redirigir a través de l'estructura d'Internet.

LISP representa una de les millors solucions existents per per la Mobilitat de Màquines Virtuals(Virtual Machine Mobility)[2] i s'utilitza per a multitud de fins, com per exemple per l'establiment de Data Centers distribuïts.[3][4]

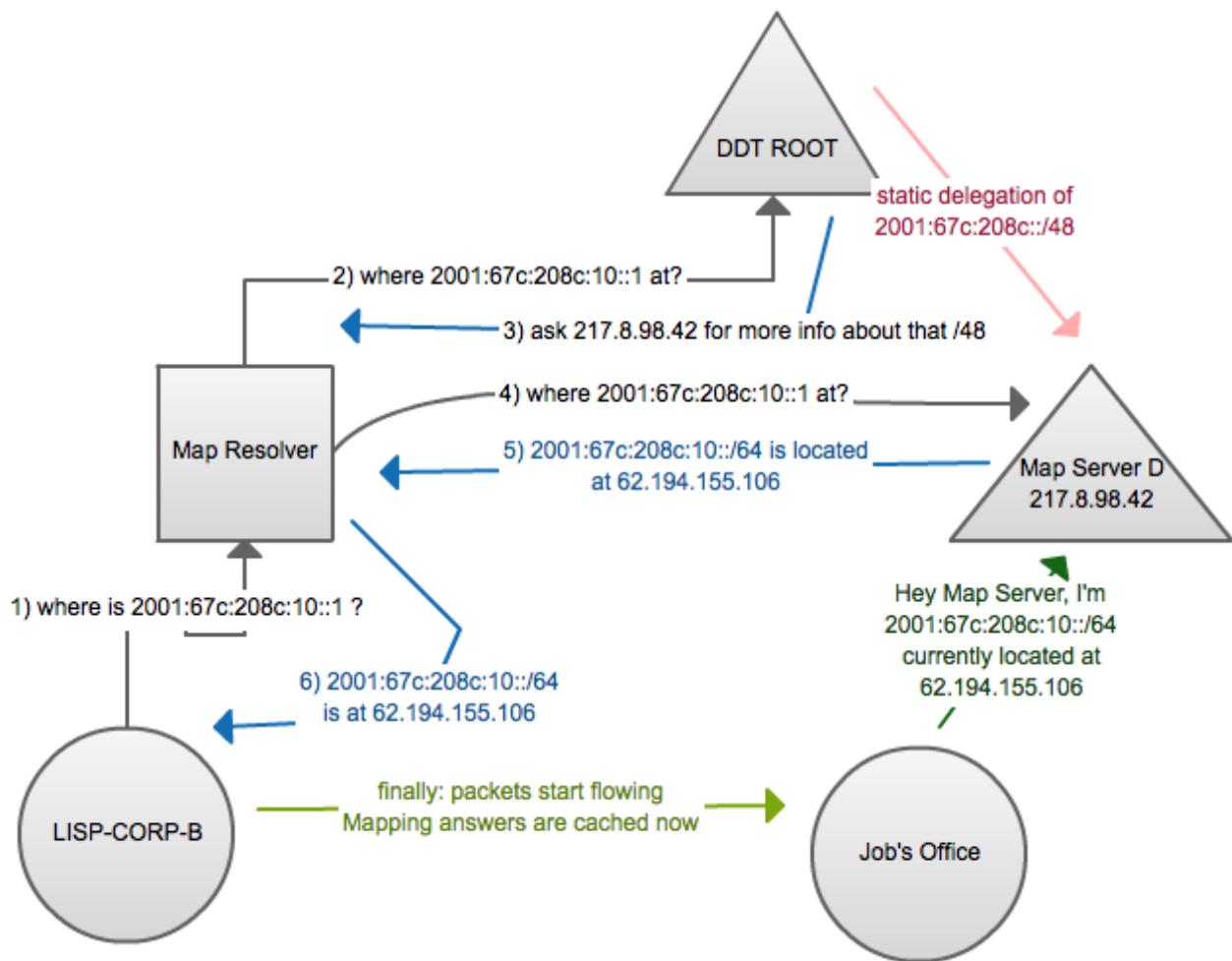
Per al funcionament de LISP, és necessària una manera de «mapejar» entre EIDs i RLOCs, i aquí és on entra LISP-DDT.

Què és LISP-DDT?

LISP-DDT[1] és una arquitectura per a una base de dades de mapeigs EID-RLOC, amb èmfasi en una alta escalabilitat. Es tracta d'una base de dades jeràrquica que representa la delegació d'autoritat per a proporcionar mapeigs. També proporciona informació de delegació als Map-Resolvers per a localitzar mapeigs EID-RLOC. Un Map-Resolver segueix el camí a través dels nodes fins que arriba al node autoritari del mapeig que busca. Es tracta d'un funcionament similar al protocol DNS, però amb la diferència de que l'a informació s'actualitza de manera dinàmica.[5]

En la següent figura, es pot veure un exemple del funcionament de LISP-DDT.

En aquest cas, la màquina «LISP-CORP-B» vol enviar dades a la màquina «Job's Office», i en coneix el seu EID(2001:67c:208c:10::1). Com que necessita saber el seu RLOC, el pregunta al seu Map Resolver. Aquest no té la informació, així que pregunta a DDT ROOT, i tot seguit segueix els referrals fins a arribar al Map Server on «Job's Office» registra el seu RLOC sempre que es connecta. Un cop el Map Resolver, obté la informació, la retorna a «LISP-CORP-B»(en realitat, normalment l'envia el Map Server, però això és per simplificar l'exemple), el qual ara ja té la informació que necessita per començar a enviar dades a «Job's Office».

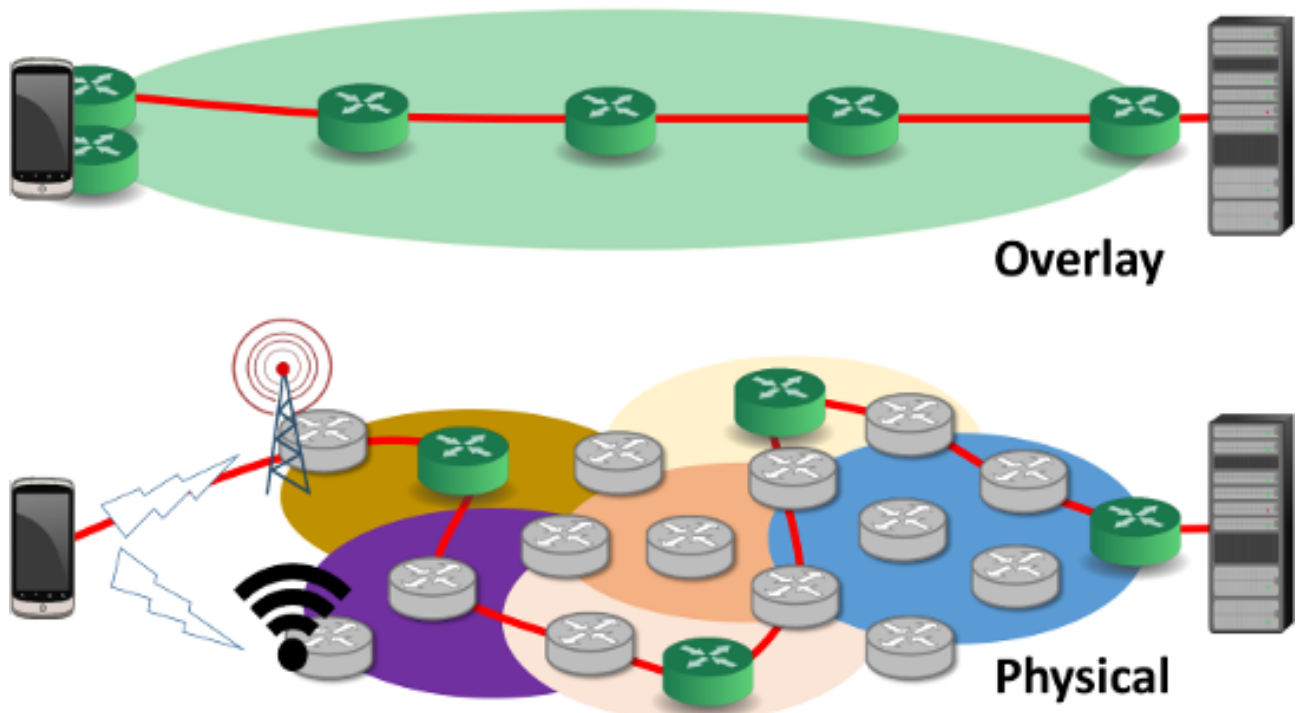


[5]

LISP-CORP-B pregunta al Map Resolver la ubicació de «Job's Office», i aquest pregunta a DDT ROOT i segueix una cadena de referrals(molt curta en aquest cas) per arribar a la resposta.

Què és OpenOverlayRouter?

Open Overlay Router[6] és una implementació de codi obert per a crear xarxes de superposició programables. Aquesta està escrita en C, i disponible per a Linux, Android, i OpenWRT.



Una xarxa de superposició comparada amb la seva xarxa física.

1.1.2 Actors implicats

El projecte va dirigit a qualsevol usuari que vulgui utilitzar Open Overlay Router juntament amb el protocol LISP-DDT.

Els actors implicats en aquest projecte son:

- Desenvolupador del projecte: Les tasques de cap de projecte, dissenyador, programador i responsable de proves han estat exercides per l'autor d'aquest projecte.
- Director del projecte: El director del projecte és Albert Cabellos Aparicio, amb Jordi Paillisé Vilanova com a co-director. El seu paper ha estat el de guiar i supervisar el compliment del calendari i objectius del desenvolupament del projecte.
- Consultor: Albert López Brescó ha estat el desenvolupador de l'aplicació per el projecte LISPmob. Ha actuat com a consultor i coordinador al llarg del projecte.
- Equip de OpenOverlayRouter: Tot el grup d'enginyers involucrats en el projecte OpenOverlayRouter es veuen beneficiats de l'èxit d'aquest projecte.
- Usuaris: OpenOverlayRouter, com a projecte de codi obert, és gratuït i obert a qualsevol usuari que en vulgui fer ús.

1.1.3 Estat de l'Art

Dins de les propostes de protocols centrats en la idea de la separació de la identitat d'un node de xarxa i la seva localització, LISP ha vist un important esforç de desenvolupament i implementació. [7]

Per a funcionar, LISP necessita un sistema de mapeig entre els EID i els RLOC.

Alguns dels proposats fins ara son:

- LISP+ALT: LISP+ALT(LISP Alternative Topology) És una arquitectura jeràrquica que fa servir BGP per a passar informació. Utilitza una xarxa superposada de BGP, a través de la qual anuncia els EID.
- LISP-DHT: LISP-DHT(LISP Distributed Hash Table) és una solució P2P(peer to peer) que guarda els enllaços entre EIDs i RLOCs en un overlay de tipus Chord(taula de hash), on cada router actua com un node del overlay.
- LISP-CONS: LISP-CONS(LISP Content distribution Overlay Network Service) és un sistema jeràrquic de distribució de contingut per mapeigs EID-RLOC. És una generalització de LISP+ALT que no utilitza el protocol d'enrutament BGP.
- LISP-TREE: LISP-TREE és un sistema jeràrquic de mapeig amb una clara separació entre l'emmagatzematge dels mapeigs i la seva obtenció. La responsabilitat de l'emmagatzematge està sota els enrutadors túnel de sortida(ETRs), mentre que el mecanisme d'obtenció està construït amb el protocol DNS.

LISP-TREE és el que va assentar les bases per LISP-DDT, el qual s'ha convertit en el sistema de mapeig més prometedor, i adoptat, per exemple, per la LISP Beta Network.[5]

En l'actualitat, existeix una implementació de LISP-DDT, però aquesta és **propietat de Cisco Systems**[5] i, per tant, no resultaria possible utilitzar-la en un projecte de codi obert com OpenOverlayRouter.

1.2 Objectius i abast

1.2.1 Objectius, formulació del problema

El projecte consisteix en realitzar una implementació del protocol LISP-DDT per a fer-lo funcionar sobre la plataforma OpenOverlayRouter.

El «problema», fent servir terminologia del TFG, és que OpenOverlayRouter no compta amb l'implementació de LISP-DDT. Com ja es mencionava en l'anterior apartat, l'única implementació que existeix actualment de LISP-DDT és una que està protegida sota propietat de Cisco Systems, i això significa que no es podria utilitzar per a OpenOverlayRouter, que és un projecte de codi obert. Per tant, degut a l'absència d'alternatives, l'única solució és implementar-lo.

1.2.2 Abast

L'abast del projecte es pot concretar en uns punts concrets, el primer dels quals és realitzar la implementació de les tres entitats que s'utilitzen en LISP-DDT: DDT-Node, DDT Map Server i DDT Map Resolver. El segon punt, molt lligat amb el primer, és realitzar les modificacions necessàries en la resta de fitxers, per tal de permetre la correcta configuració i funcionament de LISP-DDT dins de OpenOverlayRouter, sense interferir en les funcionalitats ja existents.

1.3 Metodologia i rigor

1.3.1 Metodologia

La metodologia utilitzada per al desenvolupament d'aquest projecte és una metodologia àgil basada en Scrum, consistent en períodes d'una o dues setmanes, en els que s'estableixen una sèrie d'objectius, i amb una reunió al final de cada període per analitzar el compliment dels objectius i establir els del següent període.

1.3.2 Validació

La validació de l'èxit del projecte es durà a terme a través de testos sobre les diverses funcionalitats que cada una de les entitats de LISP-DDT, llistades en l'apartat anterior, ha de ser capaç de realitzar. Aquests testos es realitzaran tant en una xarxa tancada, com en una xarxa real, en el cas del DDT Map Resolver, i es documentaran a través de captures de pantalla amb el programa WireShark, que permet veure amb detall els missatges entrants i sortints en una xarxa.

Aquestes proves es poden veure amb més detall a la [secció 5](#), «Resultats».

La metodologia a seguir encara no està completament decidida en el moment de fer aquest lliurament, però el que resulta més probable és un sistema similar a Scrum, amb períodes de una o dues setmanes en els que s'estableixin una sèrie d'objectius, i amb una reunió al final de cada període per analitzar el compliment dels objectius i establir els del següent període.

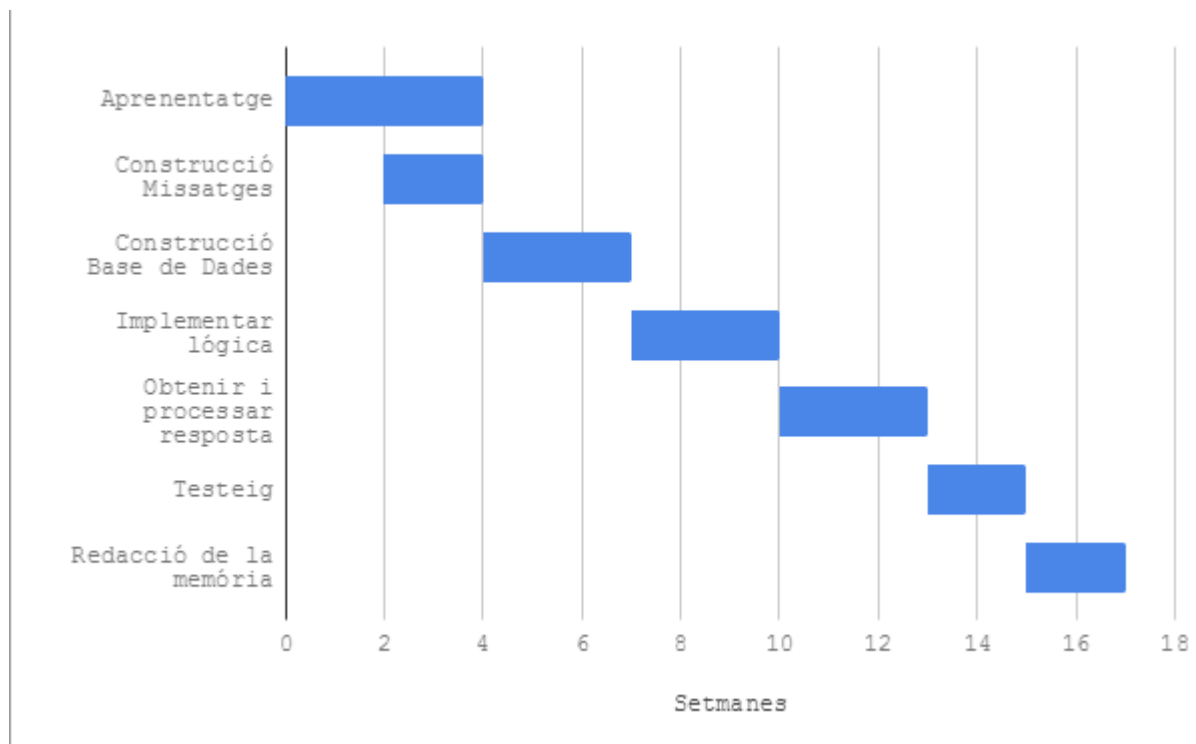
El mètode per validar l'èxit del projecte serà realitzar testos sobre la implementació realitzada, comprovant que és capaç de fer totes, o almenys la gran majoria, de les funcions que requereix per a funcionar correctament.

2 Planificació temporal

2.1 Duració del projecte

El projecte té una duració d'unes 18 setmanes, començant a comptar la setmana del 18 de Setembre, fins la setmana del 15 de Gener de 2018, tot i que les últimes setmanes estarien majorment dedicades a la redacció de la memòria i preparació de la defensa oral.

2.2 Diagrama Gantt



2.3 Tasques del projecte

Aprenentatge: Consisteix en estudiar a consciència la documentació existent sobre els temes rellevants del projecte, a fi de reduir la necessitat de futures consultes durant la realització d'aquest, i minimitzar el temps requerit per futures consultes. El temps de dedicació és d'unes 55-65 hores.

Construcció Missatges: Consisteix en analitzar l'estructura dels missatges DDT, per tal de poder construir-los de manera correcta i eficient amb el software que es crearà com a objectiu del projecte. El temps és d'unes 30 hores.

Construcció Base de Dades: Consisteix en establir i construir el sistema de dades que guardarà les dades que necessitin un cert grau de persistència per al correcte funcionament del software del projecte. El temps és de 60-70 hores.

Implementar lògica: Consisteix en analitzar la lògica dels tipus de nodes DDT a implementar, i construir la lògica software que permeti prendre automàticament les decisions segons les diferents situacions que poden passar. El temps és de 60-70 hores.

Obtenir i processar resposta: Consisteix en construir el sistema que sigui capaç de llegir els missatges DDT que es rebin, podent així obtenir-ne el tipus i tota la informació que sigui necessària, per tal d'aplicar la lògica prèviament implementada. El temps de dedicació és de 60-70 hores.

Testeig: Consisteix en provar que el software implementat funciona de manera adequada, trobant i corregint els possibles errors que hi hagi, a més d'optimitzar les parts que siguin oportunes, si es pot.

Redacció de la memòria: Consisteix en redactar el document de la Memòria del TFG, a més de la preparació per a la defensa del projecte.

2.4 Alternatives i pla d'acció

Tot i que el temps que està previst per a les diferents tasques hauria de ser suficient, sempre existeix la possibilitat que diversos imprevistos puguin entorpir el desenvolupament del projecte.

Tal com es mencionava en la secció anterior, la metodologia utilitzada consisteix en un sistema de fites setmanals basat en Scrum. Això permet que, en cas de que els objectius d'una setmana no es complissin, el que faltés d'aquells objectius es podrien traslladar a la setmana següent, adaptant, si escaigués, els objectius d'aquella setmana, o, depenent de les circumstàncies, augmentar les hores de treball de la setmana per tal de fer front a la feina addicional.

També es contempla la possibilitat d'afegir objectius addicionals a una setmana si la feina d'aquesta es realitza amb molt menys temps del previst, avançant objectius de setmanes posteriors, i obtenint així un cert marge per a cobrir futurs imprevistos.

3 Gestió econòmica i sostenibilitat

3.1 Identificació i estimació dels costos

3.1.1 Recursos humans

A l'apartat de la planificació temporal es descriu que la duració del projecte és de 18 setmanes, la qual cosa significa uns 126 dies. Si s'aproxima un volum de treball diari d'unes 3 hores, la duració total esdevé 378 hores.

Aquestes estarien dividides de la següent manera:

Rol	Hores	Salari/h	Total (€)
Director de projecte	80	17,29	1383
Enginyer de software	70	13,22	925
Programador	228	11,37	2592
Total	378	-	4900

Salari extrets de la pàgina <http://tusalarario.es/>

Concretament: [Director de Projecte](#), [Enginyer de Software](#), [Programador](#)

3.1.2 Recursos hardware

Als recursos hardware se'ls estima una vida útil d'uns 4 anys. Per establir el cost d'amortització per hora es consideren els dies hàbils de l'any(cal descomptar festius i caps de setmana), que son uns 250, i 8 hores per dia.

Hardware	Preu (€)	Hores estimades	Preu d'amortització/h (€)	Total
Ordinador portàtil Acer Aspire V15	1047	300	0,131	39,3
Total estimat	-	-	-	39,3

3.1.3 Recursos software

Els recursos software utilitzats per al projecte son gratuïts, així que aquest cost és 0.

3.1.4 Cost total

Recurs	Cost estimat (€)
Recursos humans	4900
Recursos hardware	39,3
Recursos software	0
Total	4939,3

3.2 Control de gestió

El pressupost previst en principi hauria de ser suficient per a la realització del projecte.

No obstant, sempre pot haver-hi despeses que no estiguessin previstes o que no s'haguessin contemplat al principi del projecte. Per fer front a aquestes, atès que el cost del projecte no és (relativament) gaire elevat, es podria sobreestimar el pressupost a uns 6000€, per així tenir un cert marge, la part del qual no s'acabés utilitzant quedaria com a superàvit del projecte.

3.3 Sostenibilitat i compromís social

Social

L'eina OpenOverlayRouter pot ser utilitzada tant per empreses com per particulars per a fer servir el protocol LISP. La inclusió del protocol LISP-DDT a les seves funcionalitats servirà per a millorar l'eina i oferir un millor servei als usuaris.

Ambiental

Tractant-se d'un projecte completament de software, no té cap impacte mediambiental, a banda dels indirectes causats pel hardware utilitzat, i l'electricitat consumida.

Taula

Sostenibilitat	Ambiental	Econòmica	Social
Puntuació	8	6	8

4 Desenvolupament del projecte

Les diferents parts del projecte, concretament les entitats de LISP-DDT, s'han dissenyat i desenvolupat d'acord amb el RFC 8111(<https://tools.ietf.org/html/rfc8111>)[1], document en el que s'especifica el funcionament de LISP-DDT i de les diverses entitats que aquest comprén.

Sempre que ha estat possible, s'ha seguit completament el que estava indicat en el document en qüestió. No obstant, hi ha hagut casos en que algunes coses no estaven especificades prou concretament, i ha calgut decidir la millor manera de fer-ho. També hi ha hagut casos en que s'ha fet canvis que, tot i contradir les indicacions del document, s'ha considerat que suposaven una millora, almenys en l'estat actual de LISP-DDT.

A continuació explicaré aquests casos, començant per els del primer tipus.

4.1 Decisions de disseny

S'ha creat una estructura de mapping record(una estructura de missatges utilitzada dins de diversos dels missatges de LISP) especial per al Map Referral(utilitzat per DDT) en lloc de modificar la ja existent. Això s'ha fet per evitar comportaments imprevisibles si algun nou tipus de missatge també fes servir els nous camps d'aquest per un altre fi, o en cas de que arribi un missatge mal configurat.

En un Map Resolver, la llista de nonces(informació de seguretat dels requests enviats) associada a un Pending Request es neteja cada cop que es canvia la referència utilitzada. Això significa que sempre s'acceptarà qualsevol resposta que arribi d'un dels nodes als que hem preguntat fent servir la referència actual, mentre que respostes de referències anteriors ja no interferiran.

4.2 Canvis respecte al RFC 8111

Segons l'especificació, un DDT Map Server també hauria de tenir les funcionalitats d'un DDT-Node(podent, per tant, actuar com ambdues entitats), però hem decidit no incloure-les. Això s'ha decidit així perquè en un context realista aquestes no es farien servir mai, atès que quan es configura un DDT Map Server sempre es fa només amb les funcionalitats de Map Server en ment.

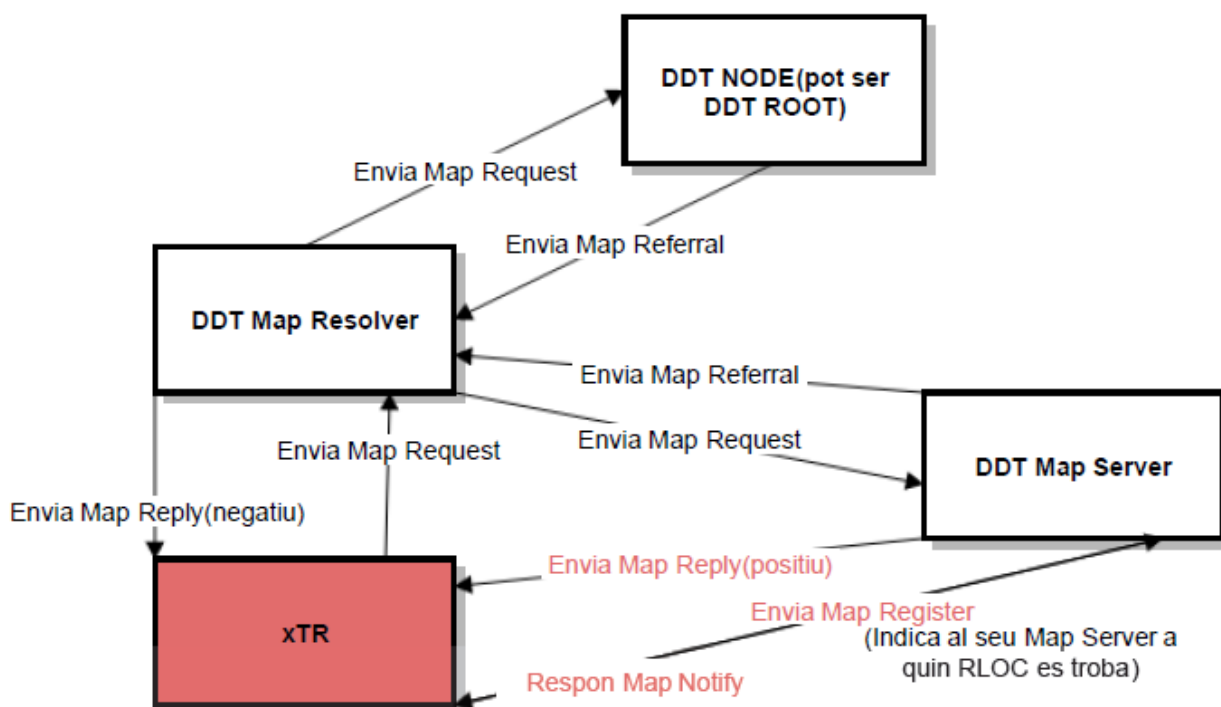
El document sembla suggerir que, en un Map Resolver, la direcció de DDT-Root es guarda com una entrada més de cache. Això suposaria problemes, perquè caldria guardar-hi un mapeig per cada possible valor del IID. En lloc d'això, es guarda apart, i s'hi accedeix si no es troba cap entrada a la cache que serveixi per el request rebut. En l'actualitat, majorment es fa servir només el IID 0, així que en un principi no hauria de suposar molt problema, però en un futur es pot esperar que se n'utilitzaran més, i el mètode proposat és millor en quant a escalabilitat que l'alternativa de guardar múltiples mapejos.

En el document s'indica que els Map Referral de tipus MS Ack(que tenen peers) s'han de guardar a la cache d'un Map Resolver. En la nostra implementació, hem decidit no fer-ho, atès que el referral contindrà la mateixa informació que el MS Referral que s'havia seguit per aconseguir-lo, però amb un problema. En l'actualitat, un grup de Peers de Map Servers no estan sincronitzats entre ells. Això

significa que el prefix que es demana podria estar configurat en alguns d'ells, i no en els altres. Amb aquesta situació, es fa necessari preguntar un per un als nodes delegats fins a rebre un MS Ack, que ja és exactament el que es feia amb el MS Referral, per tant, no fa cap falta guardar el MS Ack. No obstant, en el codi on es tracta això, he deixat unes parts en «comentari», que es poden descomentar per tal de fer que es guardin i es segueixi el comportament indicat al RFC 8111. Això s'hauria de fer si en un futur existeix aquesta sincronització entre Peers de Map Servers.

4.3 Esquema de classes

A continuació hi ha un esquema on es poden veure les classes involucrades en el projecte i la forma com interactuen entre elles. Les classes i interaccions en color vermell no han estat desenvolupades dins aquest projecte, però les he inclòs al considerar-les rellevants.



5 Resultats

El funcionament de les tres entitats de LISP-DDT és el correcte a data de redacció d'aquesta memòria. A continuació veurem en detall quines han estat les proves realitzades, separades per l'entitat LISP-DDT a la que corresponen, i després veurem la realització detallada de cada una.

Per a cada una de les entitats, s'ha provat que el seu funcionament és el correcte en totes les situacions descrites en el document RFC 8111(<https://tools.ietf.org/html/rfc8111>)[1], creant una certa configuració que permetés provar-les totes, i enviant missatges de manera manual per a desencadenar cada una de les situacions. En les màquines on es feien les proves, s'estava fent funcionar el programa WireShark, que permet veure els missatges entrants i sortints, podent així comprovar que son els correctes.

5.1 Proves(llista)

DDT-NODE:

Prova 1: Rebre un map request pel que el node no és autoritatiu; retornar `not_authoritative`

Prova 2: Rebre un map request pel que és autoritatiu, i el referral no és de map-server; retornar `node_referral`

Prova 3: Rebre un map request pel que és autoritatiu, i el referral és de map-server; retornar `ms_referral`

Prova 4: Rebre un map request pel que és autoritatiu, però no hi ha delegació configurada: retornar `delegation_hole`

DDT Map Server:

Prova 5: Rebre un map request pel que no és autoritatiu: retornar `not_authoritative`

Prova 6: Rebre un map request pel que és autoritatiu i la delegació té peers; enviar `ms_ack` amb bit d'incomplete a 0 i la llista de peers

Prova 7: Rebre un map request pel que és autoritatiu i la delegació no té peers; enviar `ms_ack` amb bit d'incomplete a 1

Prova 8: Rebre un map request pel que és autoritatiu però no hi ha site registrat; enviar `ms_not_registered`

DDT Map Resolver:

Prova 9: El Map Resolver pot seguir una cadena de referrals fins a arribar a un Map Server, i enviar-li el Map Request amb l'informació de seguretat original.

Prova 10: El Map Resolver guarda els referrals rebuts a la cache, pel que pot començar a preguntar a un node que no és root, si té la informació necessària.

Prova 11: El Map Resolver, en rebre una resposta not_authoritative, pregunta a Root(si no ho havia fet ja) o elimina el pending request i envia un map reply negatiu a qui havia fet la pregunta.

Prova 12: El Map Resolver és capaç de seguir cadenes de referrals quan està configurat amb DDT-Root real, especificat a <https://ddt-root.org>

5.2 Proves(detall)

5.2.1 Proves del DDT-NODE

La configuració del DDT-NODE és la següent:

Autoritativ per: 10.0.0.0/8, 20.0.0.0/8, 40.0.0.0/8

Té delegacions per: 10.0.0.0/8, 40.0.0.0/8(map server)

Prova 1:

El node rep un request preguntant per 30.0.0.0, per el qual no és autoritativ. El comportament esperat és un Map Referral de tipus Not Authoritative. Es pot veure a la Figura 1.

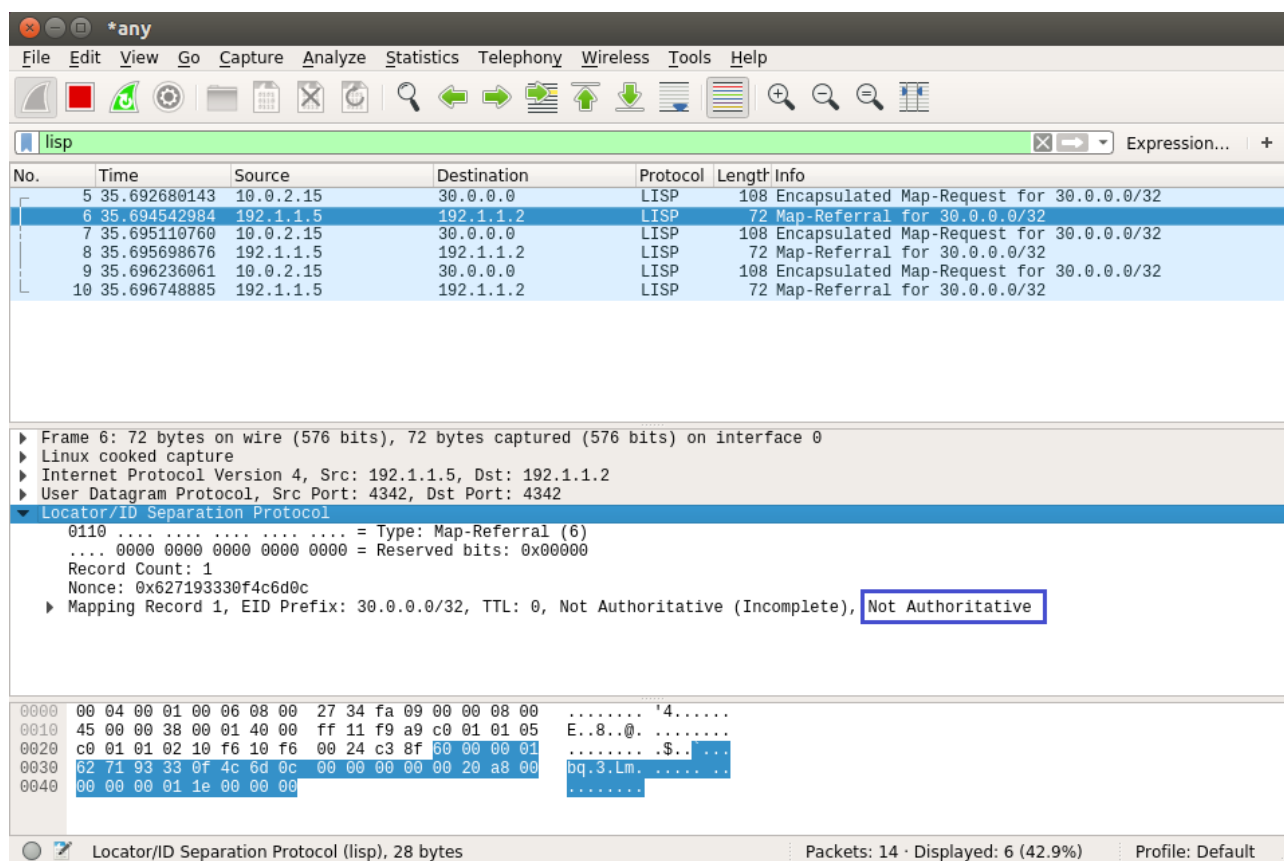


Figura 1

Prova 2:

El node rep un request preguntant per 10.0.0.0, per el qual és autoritatiu, i té una delegació que no és de tipus map-server. El comportament esperat és un Map Referral de tipus Node Referral. Es pot veure a la Figura 2.

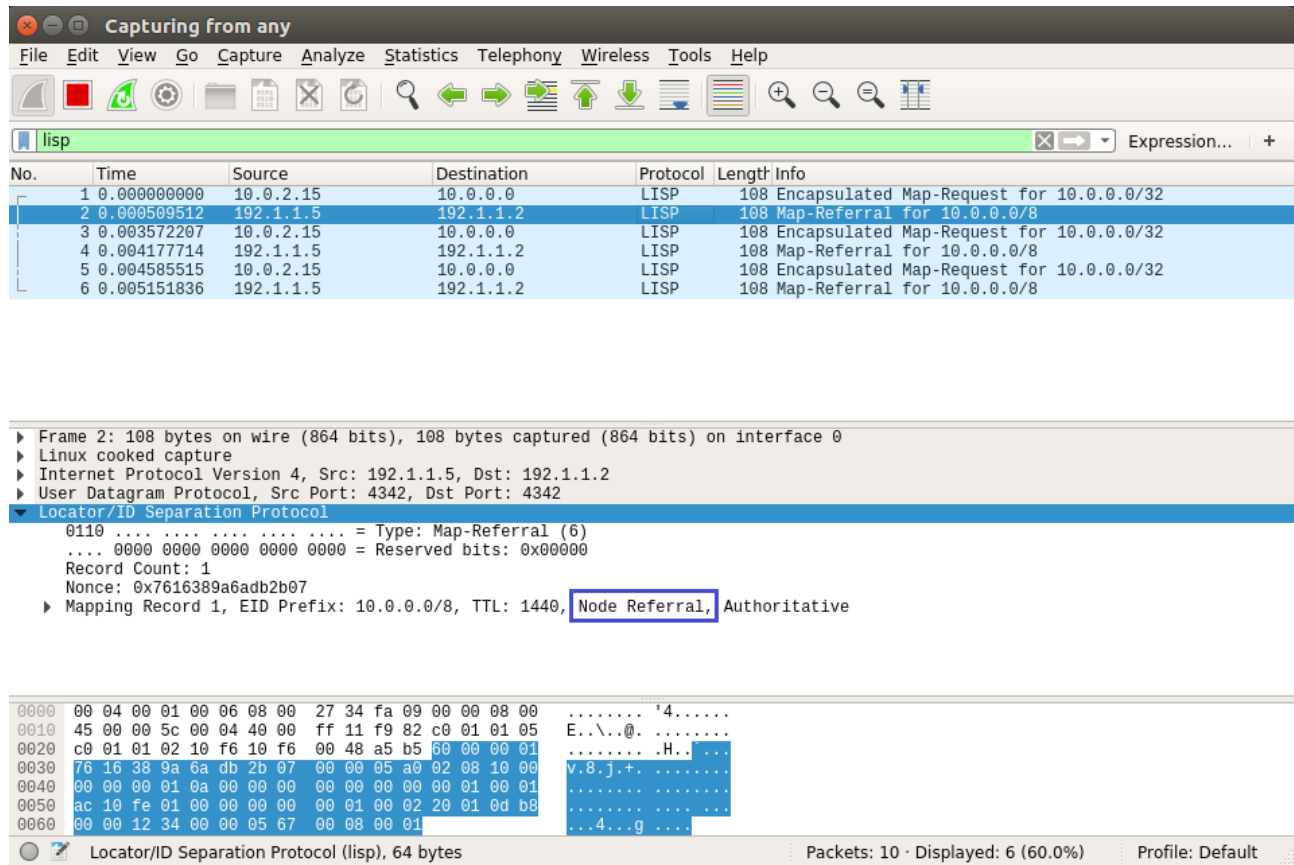


Figura 2

Prova 3:

El node rep un request preguntant per 40.0.0.0, per el qual és autoritatiu, i té una delegació que és de tipus map-server. El comportament esperat és un Map Referral de tipus MS Referral. Es pot veure a la Figura 3.

The image shows a Wireshark packet capture of LISP traffic. The top pane displays a list of six packets. Packet 2 is selected, showing details in the middle pane and a hex/ASCII dump in the bottom pane.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	40.0.0.0	LISP	108	Encapsulated Map-Request for 40.0.0.0/32
2	0.000479395	192.1.1.5	192.1.1.2	LISP	108	Map-Referral for 40.0.0.0/8
3	0.001527438	10.0.2.15	40.0.0.0	LISP	108	Encapsulated Map-Request for 40.0.0.0/32
4	0.002191308	192.1.1.5	192.1.1.2	LISP	108	Map-Referral for 40.0.0.0/8
5	0.002692665	10.0.2.15	40.0.0.0	LISP	108	Encapsulated Map-Request for 40.0.0.0/32
6	0.003230118	192.1.1.5	192.1.1.2	LISP	108	Map-Referral for 40.0.0.0/8

Frame 2: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface 0
Linux cooked capture
Internet Protocol Version 4, Src: 192.1.1.5, Dst: 192.1.1.2
User Datagram Protocol, Src Port: 4342, Dst Port: 4342
Locator/ID Separation Protocol
0110 = Type: Map-Referral (6)
.... 0000 0000 0000 0000 = Reserved bits: 0x000000
Record Count: 1
Nonce: 0x31edf9712fbe2302
Mapping Record 1, EID Prefix: 40.0.0.0/8, TTL: 1440, **Map-Server Referral**, Authoritative

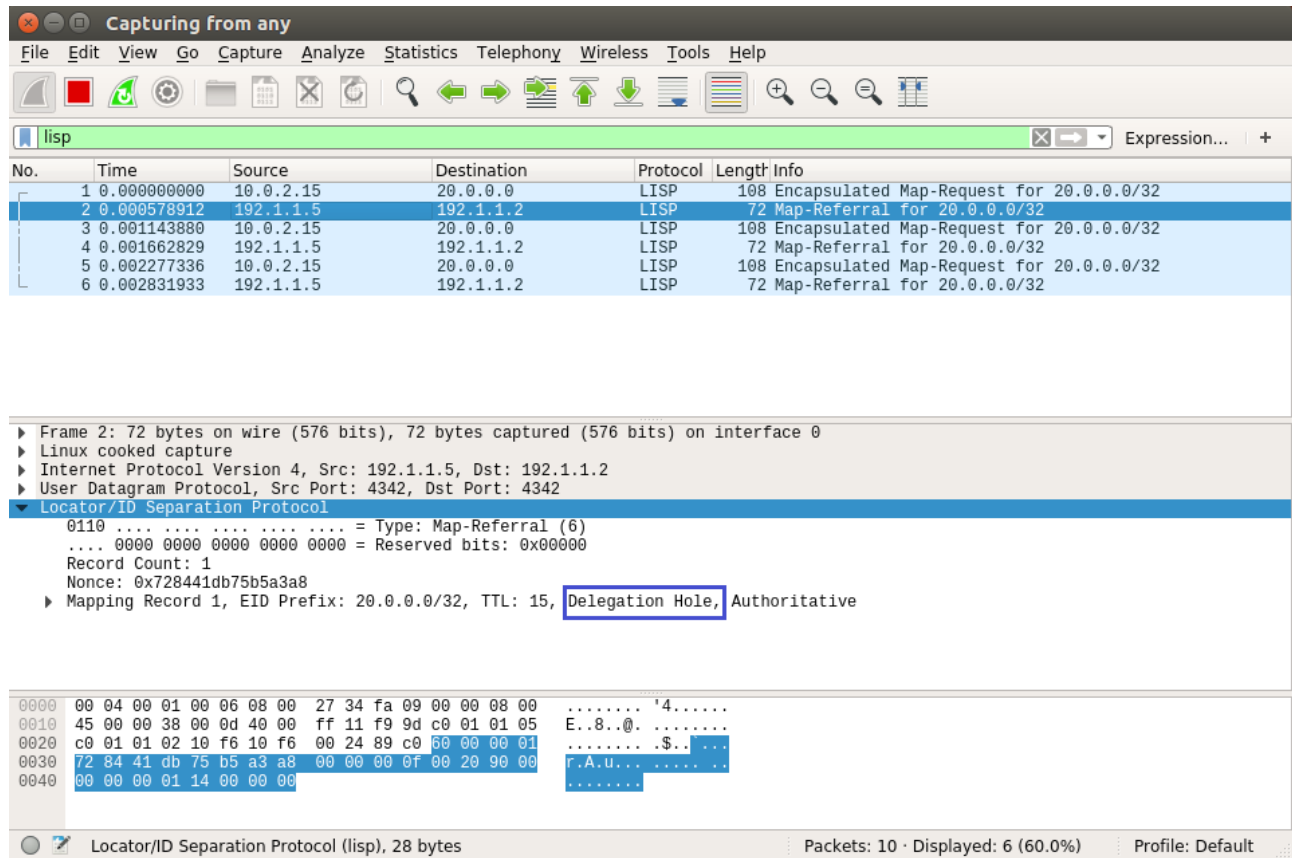
0000 00 04 00 01 00 06 08 00 27 34 fa 09 00 00 08 00 '4.....
0010 45 00 00 5c 00 07 40 00 ff 11 f9 7f c0 01 01 05 E..\\..@.
0020 c0 01 01 02 10 f6 10 f6 00 48 2e 29 00 00 00 01H.) ...
0030 31 ed f9 71 2f be 23 02 00 00 05 a0 02 08 30 00 1..q/.#.0.
0040 00 00 00 01 28 00 00 00 00 00 00 00 00 01 00 01(.....
0050 ac 10 fe 01 00 00 00 00 00 01 00 02 20 01 0d b8
0060 00 00 12 34 00 00 05 67 00 08 00 014..g

Locator/ID Separation Protocol (lisp), 64 bytes Packets: 10 · Displayed: 6 (60.0%) Profile: Default

Figura 3

Prova 4:

El node rep un request preguntant per 20.0.0.0, per el qual és autoritatiu, però no té cap delegació configurada. El comportament esperat és un Map Referral de tipus Delegation Hole. Es pot veure a la Figura 4.



The image shows a Wireshark packet capture window titled "Capturing from any". The filter bar shows "lisp". The packet list displays six packets:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	20.0.0.0	LISP	108	Encapsulated Map-Request for 20.0.0.0/32
2	0.000578912	192.1.1.5	192.1.1.2	LISP	72	Map-Referral for 20.0.0.0/32
3	0.001143880	10.0.2.15	20.0.0.0	LISP	108	Encapsulated Map-Request for 20.0.0.0/32
4	0.001662829	192.1.1.5	192.1.1.2	LISP	72	Map-Referral for 20.0.0.0/32
5	0.002277336	10.0.2.15	20.0.0.0	LISP	108	Encapsulated Map-Request for 20.0.0.0/32
6	0.002831933	192.1.1.5	192.1.1.2	LISP	72	Map-Referral for 20.0.0.0/32

The packet details pane for packet 2 shows the following structure:

- Frame 2: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0
- Linux cooked capture
- Internet Protocol Version 4, Src: 192.1.1.5, Dst: 192.1.1.2
- User Datagram Protocol, Src Port: 4342, Dst Port: 4342
- Locator/ID Separation Protocol
 - 0110 = Type: Map-Referral (6)
 - 0000 0000 0000 0000 = Reserved bits: 0x000000
 - Record Count: 1
 - Nonce: 0x728441db75b5a3a8
 - Mapping Record 1, EID Prefix: 20.0.0.0/32, TTL: 15, **Delegation Hole**, Authoritative

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0000 00 04 00 01 00 06 08 00 27 34 fa 09 00 00 08 00 ..... '4.....
0010 45 00 00 38 00 0d 40 00 ff 11 f9 9d c0 01 01 05 E..8..@.....
0020 c0 01 01 02 10 f6 10 f6 00 24 89 c0 60 00 00 01 ..... $. . . .
0030 72 84 41 db 75 b5 a3 a8 00 00 00 0f 00 20 90 00 r.A.u. . . . .
0040 00 00 00 01 14 00 00 00 ..... 
```

The status bar at the bottom indicates: "Locator/ID Separation Protocol (lisp), 28 bytes", "Packets: 10 · Displayed: 6 (60.0%)", and "Profile: Default".

Figura 4

5.2.2 Proves del DDT Map Server

La configuració del DDT Map Server és la següent:

Autoritatiu per: 10.0.0.0/32(té peers), 30.0.0.0/32, 40.0.0.0/32

Sites registrats per: 10.0.0.0/32, 40.0.0.0/32

Prova 5:

El node rep un request preguntant per 30.0.0.0, per el qual no és autoritatiu. El comportament esperat és un Map Referral de tipus Not Authoritative. Es pot veure a la Figura 5.

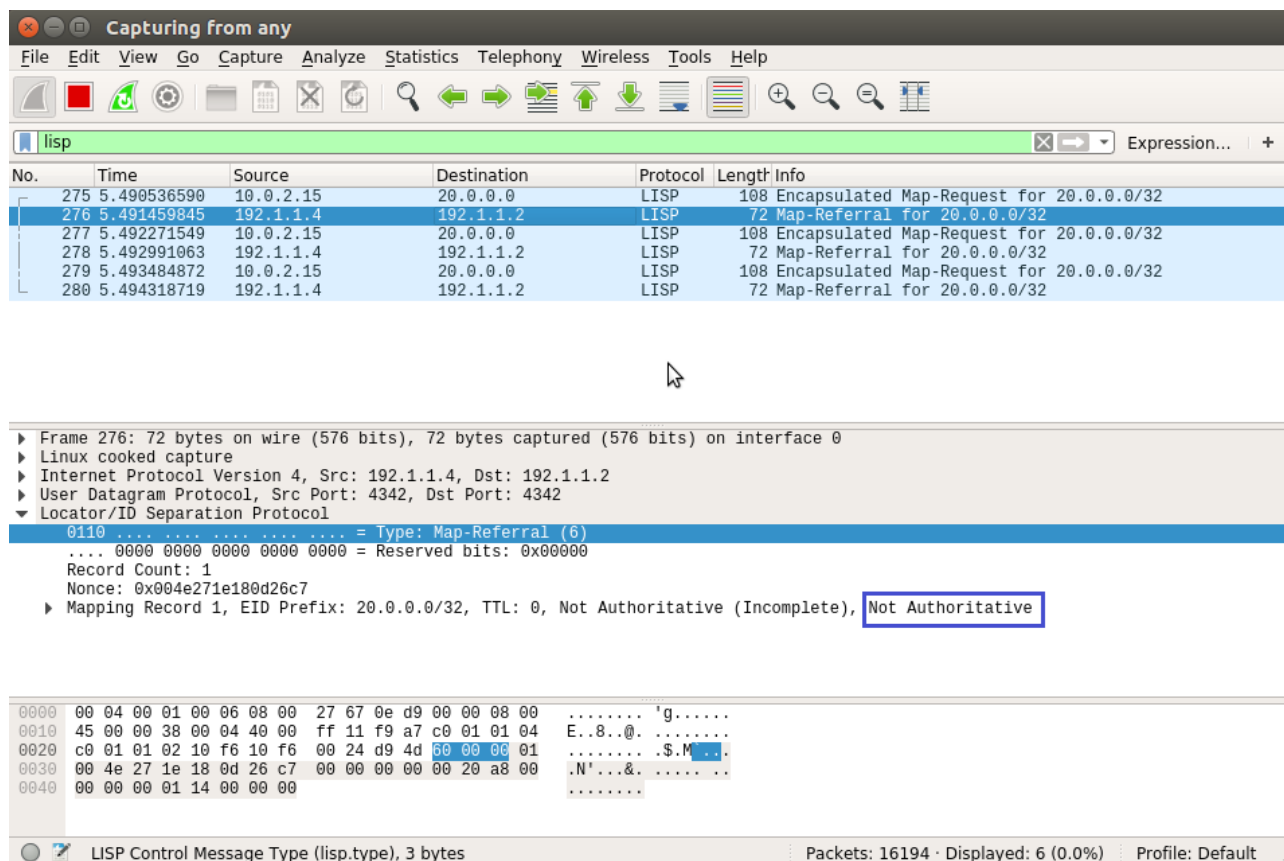


Figura 5

Prova 6:

El node rep un request preguntant per 10.0.0.0, per el qual és autoritatiu, i té una delegació amb peers. El comportament esperat és un Map Referral de tipus MS Ack, amb bit d'incomplete a 0 i una llista dels peers. El tipus del missatge es pot veure a la Figura 6, mentre que la llista de peers es pot veure a la Figura 7. Les direccions estan amagades atès que son direccions qualsevols que vaig fer servir per tal de que la prova mostrés alguna cosa, i no tenen importància per a la prova.

The image shows a Wireshark packet capture window titled "Capturing from any". The filter is set to "lisp". The packet list shows several LISP packets. Packet 2 is selected, showing details for a Map-Referral for 10.0.0.0/32. The packet is an Encapsulated Map-Request for 10.0.0.0/32, with a length of 108 bytes. The details pane shows the following information:

- Frame 2: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface 0
- Linux cooked capture
- Internet Protocol Version 4, Src: 192.1.1.4, Dst: 192.1.1.2
- User Datagram Protocol, Src Port: 4342, Dst Port: 4342
- Locator/ID Separation Protocol
- 0110 = Type: Map-Referral (6)
- 0000 0000 0000 0000 0000 = Reserved bits: 0x00000
- Record Count: 1
- Nonce: 0x2bebbb0876350790
- Mapping Record 1, EID Prefix: 10.0.0.0/32, TTL: 1440, Map-Server ACK, Authoritative

The packet bytes pane shows the raw data in hexadecimal and ASCII. The ASCII column shows the text "E..\\..@. .H. .P." and "v5.. .P.". The status bar at the bottom indicates "LISP Control Message Type (lisp.type), 3 bytes", "Packets: 15 · Displayed: 9 (60.0%)", and "Profile: Default".

Figura 6

Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

lisp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	10.0.0.0	LISP	108	Encapsulated Map-Request for 10.0.0.0/32
2	0.000511642	192.1.1.4	192.1.1.2	LISP	108	Map-Referral for 10.0.0.0/32
3	0.000636302	10.0.2.15	10.0.0.0	LISP	108	Encapsulated Map-Request for 10.0.0.0/32
4	0.001012931	10.0.2.15	10.0.0.0	LISP	108	Encapsulated Map-Request for 10.0.0.0/32
5	0.001581639	192.1.1.4	192.1.1.2	LISP	108	Map-Referral for 10.0.0.0/32
6	0.001758673	10.0.2.15	10.0.0.0	LISP	108	Encapsulated Map-Request for 10.0.0.0/32
7	0.002001744	10.0.2.15	10.0.0.0	LISP	108	Encapsulated Map-Request for 10.0.0.0/32
8	0.002265296	192.1.1.4	192.1.1.2	LISP	108	Map-Referral for 10.0.0.0/32
9	0.002373324	10.0.2.15	10.0.0.0	LISP	108	Encapsulated Map-Request for 10.0.0.0/32

Record TTL: 1440
 Locator Count: 3
 EID Mask Length: 32
 010. = Action: Send-Map-Request (2)
 ...1 = Authoritative bit: Set
 0... = Incomplete: Not set
 000 0000 0000 = Reserved: 0x000
 0000 = Signature Count: 0
 0000 0000 0000 = Mapping Version: 0
 EID Prefix AFI: IPv4 (1)
 EID Prefix: 10.0.0.0

- Locator Record 1, RLOC: [REDACTED], Reachable, Priority/Weight: 0/0, Multicast Priority/Weight: 0/0
- Locator Record 2, RLOC: [REDACTED], Reachable, Priority/Weight: 0/0, Multicast Priority/Weight: 0/0
- Locator Record 3, Local RLOC: 192.1.1.4, Reachable, Priority/Weight: 0/0, Multicast Priority/Weight: 0/0

0000 00 04 00 01 00 06 08 00 27 67 0e d9 00 00 08 00 'g.....
 0010 45 00 00 5c 00 07 40 00 ff 11 f9 80 c0 01 01 04 E..\\..@.
 0020 c0 01 01 02 10 f6 10 f6 00 48 1e b7 60 00 00 01H.. ..
 0030 2b eb bb 08 76 35 07 90 00 00 05 a0 03 20 50 00 +...v5.. P.
 0040 00 00 00 01 0a 00 00 00 00 00 00 00 01 00 01
 0050 ac 10 fe 01 00 00 00 00 00 01 00 01 ac 10 fe 02
 0060 00 00 00 00 00 05 00 01 c0 01 01 04
 LISP Control Message Type (lisp.type), 3 bytes

Packets: 15 · Displayed: 9 (60.0%) Profile: Default

Figura 7

Prova 7:

El node rep un request preguntant per 40.0.0.0, per el qual és autoritatiu, i té una delegació sense peers. El comportament esperat és un Map Referral de tipus MS Ack, amb bit d'incomplete a 1 i sense llista dels peers. El tipus del missatge es pot veure a la Figura 8, mentre que l'absència de la llista de peers es pot veure a la Figura 9, la qual es pot comparar amb la Figura 7 per a observar la diferència.

The image shows a Wireshark capture of LISP messages. The top pane displays a list of captured packets. Packet 2 is selected, showing a Map-Referral for 40.0.0.0/32. The bottom pane shows the details of this packet, including the Locator/ID Separation Protocol (LISP) header and the Mapping Record. The Mapping Record is of type 'Map-Server ACK (Incomplete)' and is authoritative. The bottom pane also shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	40.0.0.0	LISP	108	Encapsulated Map-Request for 40.0.0.0/32
2	0.000632877	192.1.1.4	192.1.1.2	LISP	72	Map-Referral for 40.0.0.0/32
3	0.000977674	10.0.2.15	40.0.0.0	LISP	108	Encapsulated Map-Request for 40.0.0.0/32
4	0.002213992	10.0.2.15	40.0.0.0	LISP	108	Encapsulated Map-Request for 40.0.0.0/32
5	0.002681178	192.1.1.4	192.1.1.2	LISP	72	Map-Referral for 40.0.0.0/32
6	0.002987775	10.0.2.15	40.0.0.0	LISP	108	Encapsulated Map-Request for 40.0.0.0/32
7	0.003447983	10.0.2.15	40.0.0.0	LISP	108	Encapsulated Map-Request for 40.0.0.0/32
8	0.003795535	192.1.1.4	192.1.1.2	LISP	72	Map-Referral for 40.0.0.0/32
9	0.004003782	10.0.2.15	40.0.0.0	LISP	108	Encapsulated Map-Request for 40.0.0.0/32

Frame 2: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0
Linux cooked capture
Internet Protocol Version 4, Src: 192.1.1.4, Dst: 192.1.1.2
User Datagram Protocol, Src Port: 4342, Dst Port: 4342
Locator/ID Separation Protocol
0110 = Type: Map-Referral (6)
.... 0000 0000 0000 0000 0000 = Reserved bits: 0x00000
Record Count: 1
Nonce: 0x4430e206227d5bef
Mapping Record 1, EID Prefix: 40.0.0.0/32, TTL: 1440, Map-Server ACK (Incomplete), Authoritative

0000 00 04 00 01 00 06 08 00 27 67 0e d9 00 00 08 00 'g.....
0010 45 00 00 38 00 0d 40 00 ff 11 f9 9e c0 01 01 04 E..8..@.
0020 c0 01 01 02 10 f6 10 f6 00 24 d1 4a 60 00 00 01 \$.J...
0030 44 30 e2 06 22 7d 5b ef 00 00 05 a0 00 20 58 00 D0.."}[. X.
0040 00 00 00 01 28 00 00 00 (...

LISP Control Message Type (lisp.type), 3 bytes Packets: 19 · Displayed: 9 (47.4%) Profile: Default

Figura 8

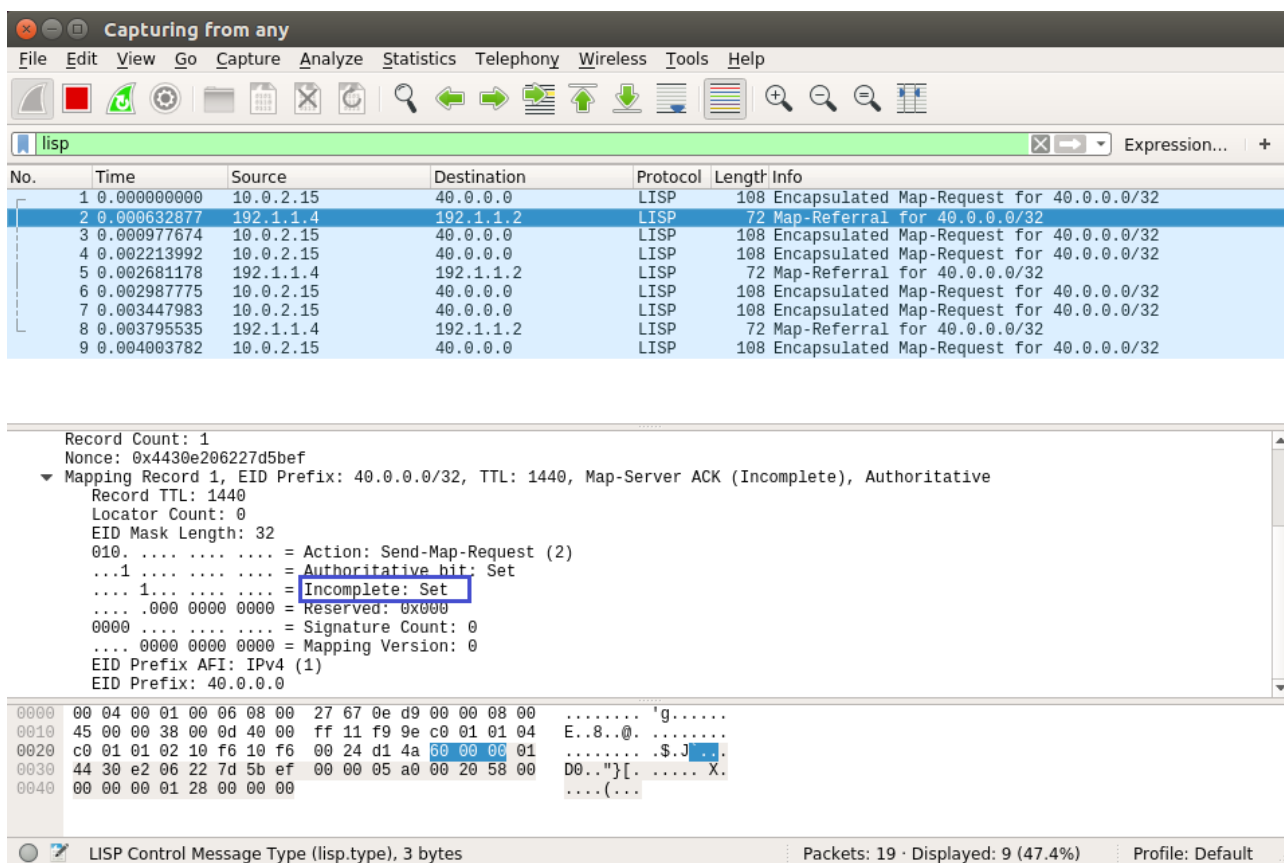


Figura 9

Prova 8:

El node rep un request preguntant per 30.0.0.0, per el qual és autoritatiu(sense peers), però no té cap registre. El comportament esperat és un Map Referral de tipus MS Not-Registered, amb bit d'incomplete a 1. Es pot veure a la Figura 10.

Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

lisp

No.	Time	Source	Destination	Protocol	Length	Info
3	3.225990993	10.0.2.15	30.0.0.0	LISP	108	Encapsulated Map-Request for 30.0.0.0/32
4	3.226776127	192.1.1.4	192.1.1.2	LISP	72	Map-Referral for 30.0.0.0/32
5	3.227152531	10.0.2.15	30.0.0.0	LISP	108	Encapsulated Map-Request for 30.0.0.0/32
6	3.227524871	192.1.1.4	192.1.1.2	LISP	72	Map-Referral for 30.0.0.0/32
7	3.227872650	10.0.2.15	30.0.0.0	LISP	108	Encapsulated Map-Request for 30.0.0.0/32
8	3.228201881	192.1.1.4	192.1.1.2	LISP	72	Map-Referral for 30.0.0.0/32

▶ Frame 4: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0
 ▶ Linux cooked capture
 ▶ Internet Protocol Version 4, Src: 192.1.1.4, Dst: 192.1.1.2
 ▶ User Datagram Protocol, Src Port: 4342, Dst Port: 4342
 ▼ Locator/ID Separation Protocol

0110 = Type: Map-Referral (6)
 0000 0000 0000 0000 = Reserved bits: 0x0000
 Record Count: 1
 Nonce: 0x40681cc65239977b
 ▶ Mapping Record 1, EID Prefix: 30.0.0.0/32, TTL: 15, Map-Server Not Registered (Incomplete), Authoritative

```

0000 00 04 00 01 00 06 08 00 27 67 0e d9 00 00 08 00 ..... 'g.....
0010 45 00 00 38 00 13 40 00 ff 11 f9 98 c0 01 01 04 E..8..@. ....
0020 c0 01 01 02 10 f6 10 f6 00 24 1e 9c 60 00 00 01 ..... $....
0030 40 68 1c c6 52 39 97 7b 00 00 00 0f 00 20 78 00 @h..R9.{ .... x.
0040 00 00 00 01 1e 00 00 00 .....
  
```

LISP Control Message Type (lisp.type), 3 bytes Packets: 12 · Displayed: 6 (50.0%) Profile: Default

Figura 10

5.2.3 Proves del DDT Map Resolver

La configuració de la xarxa interna utilitzada per les proves 9, 10 i 11 és la següent:

192.1.1.1 és el Map Resolver, té configurat com a DDT-Root el 192.1.1.5

192.1.1.5 és un DDT-Node, és autoritatiu per 10.0.0.0 i per 30.0.0.0 i ambdós els delega a 192.1.1.3

192.1.1.3 és un DDT-Node, és autoritatiu per 10.0.0.0, el qual delega a 192.1.1.4, però no és autoritatiu per 30.0.0.0

192.1.1.4 és un DDT Map Server, és autoritatiu per 10.0.0.0, per el qual té registre(amb peers)

Prova 9:

El Map Resolver rep un request preguntant per 10.0.0.0, i pregunta a DDT-Root(192.1.1.5), el qual li envia un Map Referral de tipus Node Referral indicant a 192.1.1.3. Aquest delega a 192.1.1.4, i quan el Map Resolver rep el MS Ack de 192.1.1.4, re-envia el request amb la informació de seguretat del client. Aquesta cadena es pot apreciar a la Figura 11, observant les adreces d'origen(Source) dels Map Referral que 191.1.1.1 va rebent.

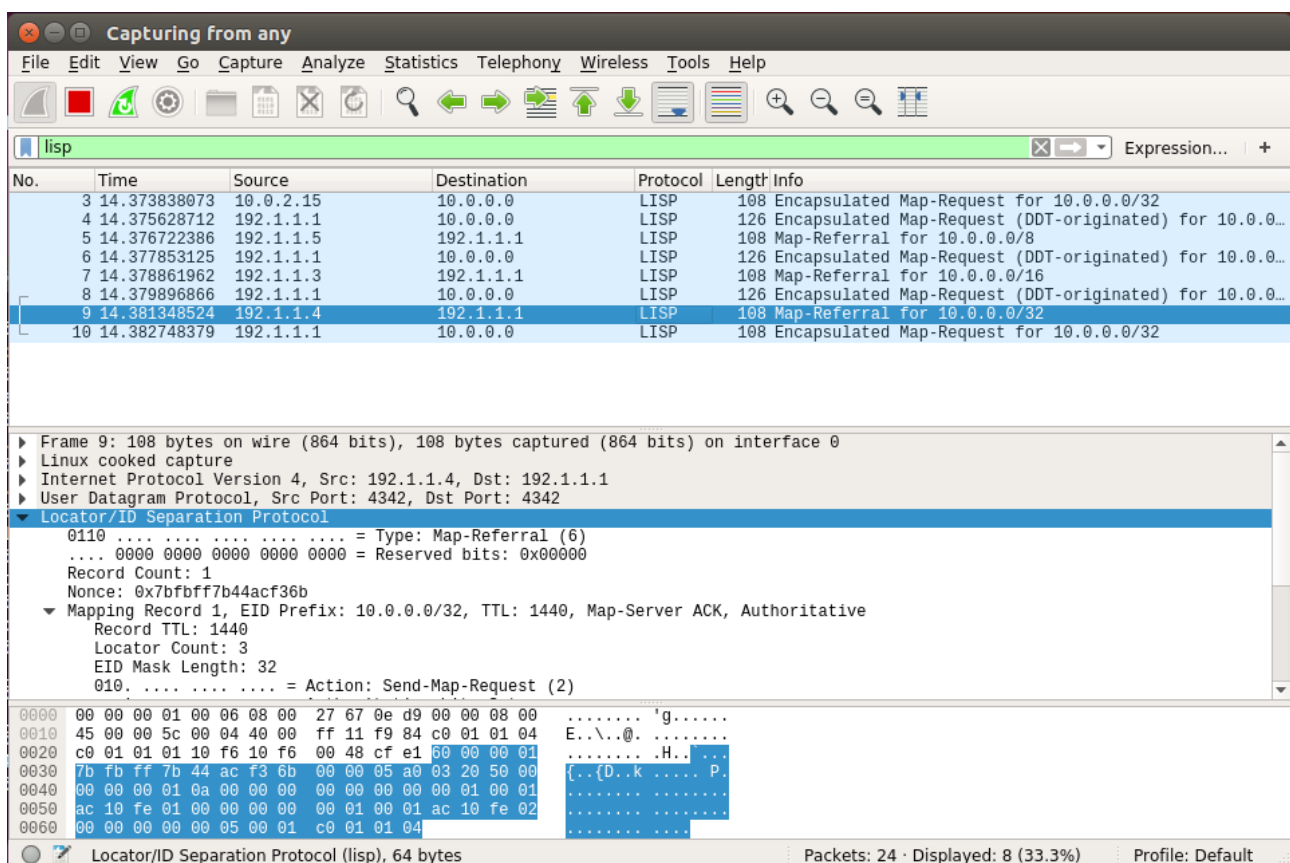


Figura 11

Prova 10:

A continuació de la prova anterior, es rep un altre request preguntant per 10.0.0.0. Com que el Map Resolver encara té informació guardada a la cache, comença preguntant directament a 192.1.1.4, el qual li retorna un MS Ack. Es pot veure a la figura 12.

The image shows a Wireshark packet capture window titled "Capturing from any". The filter bar shows "lisp". The packet list displays several packets, with packet 28 selected. The packet details pane shows the structure of the selected packet, which is a Locator/ID Separation Protocol (LISP) Map-Request. The packet is encapsulated in an Internet Protocol Version 4 (IPv4) header and a User Datagram Protocol (UDP) header. The source IP is 192.1.1.1 and the destination IP is 10.0.0.0. The packet length is 126 bytes. The packet bytes pane shows the raw data of the packet, including the LISP header and the Map-Request body.

No.	Time	Source	Destination	Protocol	Length	Info
3	14.373838073	10.0.2.15	10.0.0.0	LISP	108	Encapsulated Map-Request for 10.0.0.0/32
4	14.375628712	192.1.1.1	10.0.0.0	LISP	126	Encapsulated Map-Request (DDT-originated) for 10.0.0...
5	14.376722386	192.1.1.5	192.1.1.1	LISP	108	Map-Referral for 10.0.0.0/8
6	14.377853125	192.1.1.1	10.0.0.0	LISP	126	Encapsulated Map-Request (DDT-originated) for 10.0.0...
7	14.378861962	192.1.1.3	192.1.1.1	LISP	108	Map-Referral for 10.0.0.0/16
8	14.379896866	192.1.1.1	10.0.0.0	LISP	126	Encapsulated Map-Request (DDT-originated) for 10.0.0...
9	14.381348524	192.1.1.4	192.1.1.1	LISP	108	Map-Referral for 10.0.0.0/32
10	14.382748379	192.1.1.1	10.0.0.0	LISP	108	Encapsulated Map-Request for 10.0.0.0/32
27	157.269572696	10.0.2.15	10.0.0.0	LISP	108	Encapsulated Map-Request for 10.0.0.0/32
28	157.270873395	192.1.1.1	10.0.0.0	LISP	126	Encapsulated Map-Request (DDT-originated) for 10.0.0...
29	157.272349736	192.1.1.4	192.1.1.1	LISP	108	Map-Referral for 10.0.0.0/32
30	157.276259413	192.1.1.1	10.0.0.0	LISP	108	Encapsulated Map-Request for 10.0.0.0/32

Frame 28: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface 0

- Linux cooked capture
- Internet Protocol Version 4, Src: 192.1.1.1, Dst: 192.1.1.4
- User Datagram Protocol, Src Port: 4342, Dst Port: 4342
- Locator/ID Separation Protocol
 - 1000 = Type: Encapsulated Control Message (8)
 - 0... = S bit (LISP-SEC capable): Not set
 - 1... = D bit (DDT-originated): Set
 - 0000 0000 0000 0000 0000 0000 = Reserved bits: 0x00000000
- Internet Protocol Version 4, Src: 192.1.1.1, Dst: 10.0.0.0
- User Datagram Protocol, Src Port: 4342, Dst Port: 4342
- Locator/ID Separation Protocol
 - 0001 = Type: Map-Request (1)
 - 0000 00... = Flags: 0x00

0010 45 00 00 6e 00 0a 40 00 ff 11 f9 6c c0 01 01 01 E..n..@. ...1....

0020 c0 01 01 04 10 f6 10 f6 00 5a a2 93 84 00 00 00Z..

0030 45 00 00 4e 00 09 40 00 ff 11 b0 93 c0 01 01 01 E..N..@.

0040 0a 00 00 00 10 f6 10 f6 00 3a 7b 81 10 00 01 01: {.....

0050 da bf 7f 5f eb 2c b7 90 00 01 c0 01 01 01 00 02 ..._,,.....

0060 fd 00 00 00 00 00 00 00 00 00 00 00 00 00 01:.....

0070 00 01 c0 01 01 01 00 20 00 01 0a 00 00 00 00:.....

Locator/ID Separation Protocol (lisp), 4 bytes

Packets: 38 · Displayed: 12 (31.6%) Profile: Default

Figura 12

Prova 11:

El Map Resolver rep un request preguntant per 30.0.0.0, i pregunta a DDT-Root(192.1.1.5), el qual li envia un Map Referral de tipus Node Referral indicant a 192.1.1.3. El Map Resolver pregunta a 192.1.1.3, i aquest retorna un Map Referral de tipus Not Authoritative. Com que el Map Resolver ja havia preguntat a Root(192.1.1.5), el que fa és eliminar el request pendent i enviar un Map Reply negatiu al client(192.1.1.2). Això es pot observar a la Figura 13. A continuació, rep un altre request preguntant per 30.0.0.0. En aquest cas, té informació a la cache que apunta a 192.1.1.3, així que pregunta a aquest. Quan rep un Map Referral de tipus Not Authoritative, com que aquesta vegada no havia passat per Root(192.1.1.5), el que fa és preguntar-hi. Això es pot observar a la Figura 14 i la Figura 15.

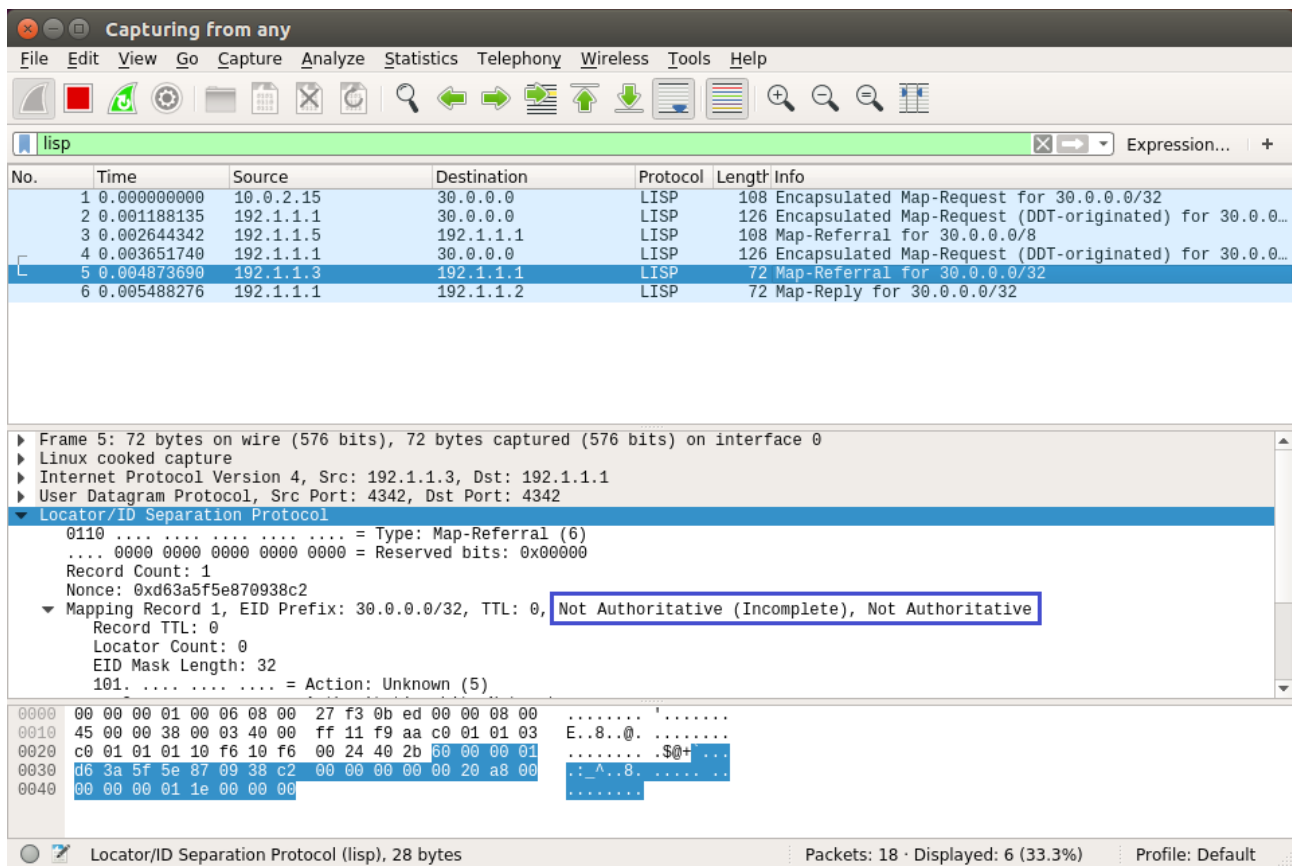


Figura 13

Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

lisp

No.	Time	Source	Destination	Protocol	Length	Info
3	0.002644342	192.1.1.5	192.1.1.1	LISP	108	Map-Referral for 30.0.0.0/8
4	0.003651740	192.1.1.1	30.0.0.0	LISP	126	Encapsulated Map-Request (DDT-originated) for 30.0...
5	0.004873690	192.1.1.3	192.1.1.1	LISP	72	Map-Referral for 30.0.0.0/32
6	0.005488276	192.1.1.1	192.1.1.2	LISP	72	Map-Reply for 30.0.0.0/32
19	87.080167488	10.0.2.15	30.0.0.0	LISP	108	Encapsulated Map-Request for 30.0.0.0/32
20	87.081082589	192.1.1.1	30.0.0.0	LISP	126	Encapsulated Map-Request (DDT-originated) for 30.0...
21	87.082166913	192.1.1.3	192.1.1.1	LISP	72	Map-Referral for 30.0.0.0/32
22	87.082750217	192.1.1.1	30.0.0.0	LISP	126	Encapsulated Map-Request (DDT-originated) for 30.0...
23	87.083623162	192.1.1.5	192.1.1.1	LISP	108	Map-Referral for 30.0.0.0/8
24	87.084818137	192.1.1.1	30.0.0.0	LISP	126	Encapsulated Map-Request (DDT-originated) for 30.0...
25	87.085832728	192.1.1.3	192.1.1.1	LISP	72	Map-Referral for 30.0.0.0/32
26	87.086142660	192.1.1.1	192.1.1.2	LISP	72	Map-Reply for 30.0.0.0/32

▶ Frame 21: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0

▶ Linux cooked capture

▶ Internet Protocol Version 4, Src: 192.1.1.3, Dst: 192.1.1.1

▶ User Datagram Protocol, Src Port: 4342, Dst Port: 4342

▼ Locator/ID Separation Protocol

0110 = Type: Map-Referral (6)

.... 0000 0000 0000 0000 = Reserved bits: 0x000000

Record Count: 1

Nonce: 0xfd6afd5ad2ab685c

▼ Mapping Record 1, EID Prefix: 30.0.0.0/32, TTL: 0, Not Authoritative (Incomplete), Not Authoritative

Record TTL: 0

Locator Count: 0

EID Mask Length: 32

101. = Action: Unknown (5)

```

0000 00 00 00 01 00 06 08 00 27 f3 0b ed 00 00 08 00 .....
0010 45 00 00 38 00 04 40 00 ff 11 f9 a9 c0 01 01 03 E..8..@.
0020 c0 01 01 01 10 f6 10 f6 00 24 ff c1 60 00 00 01 .....$.
0030 fd 6a fd 5a d2 ab 68 5c 00 00 00 00 00 20 a8 00 .j.Z..h\
0040 00 00 00 01 1e 00 00 00 .....

```

Locator/ID Separation Protocol (lisp), 28 bytes Packets: 38 · Displayed: 14 (36.8%) Profile: Default

Figura 14

Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

lisp

No.	Time	Source	Destination	Protocol	Length	Info
3	0.002644342	192.1.1.5	192.1.1.1	LISP	108	Map-Referral for 30.0.0.0/8
4	0.003651740	192.1.1.1	30.0.0.0	LISP	126	Encapsulated Map-Request (DDT-originated) for 30.0...
5	0.004873690	192.1.1.3	192.1.1.1	LISP	72	Map-Referral for 30.0.0.0/32
6	0.005488276	192.1.1.1	192.1.1.2	LISP	72	Map-Reply for 30.0.0.0/32
19	87.080167488	10.0.2.15	30.0.0.0	LISP	108	Encapsulated Map-Request for 30.0.0.0/32
20	87.081082589	192.1.1.1	30.0.0.0	LISP	126	Encapsulated Map-Request (DDT-originated) for 30.0...
21	87.082166913	192.1.1.3	192.1.1.1	LISP	72	Map-Referral for 30.0.0.0/32
22	87.082750217	192.1.1.1	30.0.0.0	LISP	126	Encapsulated Map-Request (DDT-originated) for 30.0...
23	87.083623162	192.1.1.5	192.1.1.1	LISP	108	Map-Referral for 30.0.0.0/8
24	87.084818137	192.1.1.1	30.0.0.0	LISP	126	Encapsulated Map-Request (DDT-originated) for 30.0...
25	87.085832728	192.1.1.3	192.1.1.1	LISP	72	Map-Referral for 30.0.0.0/32
26	87.086142660	192.1.1.1	192.1.1.2	LISP	72	Map-Reply for 30.0.0.0/32

▶ Frame 22: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface 0

▶ Linux cooked capture

▶ Internet Protocol Version 4, Src: 192.1.1.1, Dst: 192.1.1.5

▶ User Datagram Protocol, Src Port: 4342, Dst Port: 4342

▼ Locator/ID Separation Protocol

1000 = Type: Encapsulated Control Message (8)

..... = S bit (LISP-SEC capable): Not set

..... = D bit (DDT-originated): Set

.....0000 0000 0000 0000 0000 0000 = Reserved bits: 0x00000000

▶ Internet Protocol Version 4, Src: 192.1.1.1, Dst: 30.0.0.0

▶ User Datagram Protocol, Src Port: 4342, Dst Port: 4342

▼ Locator/ID Separation Protocol

0001 = Type: Map-Request (1)

..... = Flags: 0x00

0010 45 00 00 6e 00 15 40 00 ff 11 f9 60 c0 01 01 01 E..n..@.

0020 c0 01 01 05 10 f6 10 f6 00 5a b6 92 84 00 00 00Z..

0030 45 00 00 4e 00 14 40 00 ff 11 9c 88 c0 01 01 01 E..N..@.

0040 1e 00 00 00 10 f6 10 f6 00 3a 92 c2 10 00 01 01:.....

0050 ef 2e d5 5a d2 c3 26 4e 00 01 c0 01 01 01 00 02 ...Z..&N

0060 fd 00 00 00 00 00 00 00 00 00 00 00 00 00 01:.....

0070 00 01 c0 01 01 01 00 20 00 01 1e 00 00 00 00:.....

Locator/ID Separation Protocol (lisp), 4 bytes

Packets: 38 · Displayed: 14 (36.8%) Profile: Default

Figura 15

Prova 12:

Amb el Map Resolver configurat amb DDT-Root real(<https://ddt-root.org>), s'envien dos requests per a dues direccions LISP; una que se sap que no està configurada(Figura 16) i que per tant farà que s'aturi la cadena i s'envii un Map Reply negatiu, i una altra que se sap que sí que està correctament configurada(Figura 17). Amb aquestes consultes es pot veure que el Map Resolver és capaç de funcionar fora de la xarxa interna. Es pot saber l'estat d'adreces LISP a través de <http://nic.lisp4.net/lisp-site>

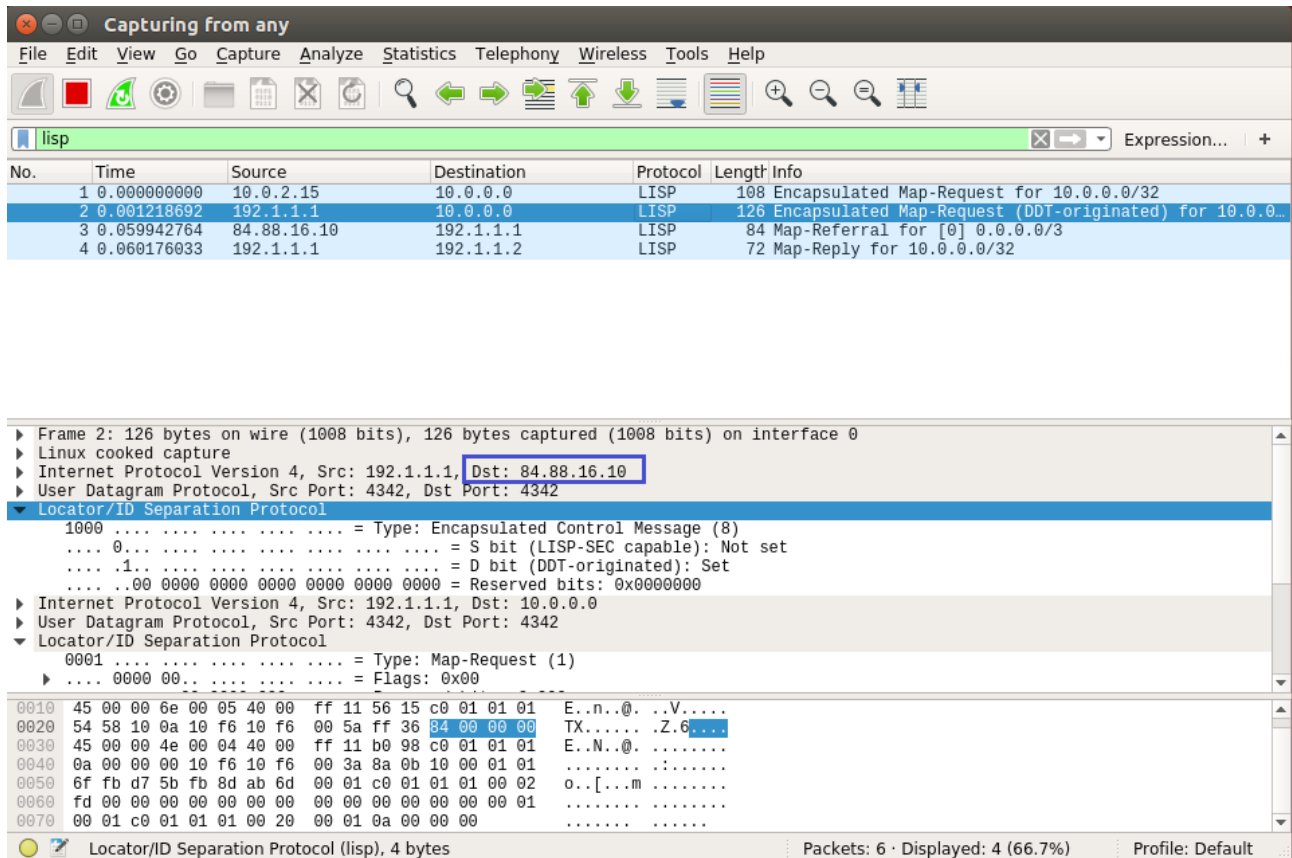


Figura 16

Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

lisp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	153.16.59.224	LISP	108	Encapsulated Map-Request for 153.16.59.224/32
2	0.000239826	192.1.1.1	153.16.59.224	LISP	126	Encapsulated Map-Request (DDT-originated) for 153.16...
3	1.199251008	84.88.16.10	192.1.1.1	LISP	120	Map-Referral for [0] 153.16.0.0/16
4	1.200546436	192.1.1.1	153.16.59.224	LISP	126	Encapsulated Map-Request (DDT-originated) for 153.16...
5	1.739784442	84.88.81.5	192.1.1.1	LISP	120	Map-Referral for [0] 153.16.32.0/19
6	1.740800160	192.1.1.1	153.16.59.224	LISP	126	Encapsulated Map-Request (DDT-originated) for 153.16...
7	2.000177022	147.83.131.32	192.1.1.1	LISP	120	Map-Referral for [0] 153.16.59.224/28
8	2.000500075	192.1.1.1	153.16.59.224	LISP	108	Encapsulated Map-Request for 153.16.59.224/32

Frame 8: 108 bytes on wire (864 bits), 108 bytes captured (864 bits) on interface 0

Linux cooked capture

Internet Protocol Version 4, Src: 192.1.1.1, Dst: 147.83.131.32

User Datagram Protocol, Src Port: 4342, Dst Port: 4342

Locator/ID Separation Protocol

1000 = Type: Encapsulated Control Message (8)

..... = S bit (LISP-SEC capable): Not set

..... = D bit (DDT-originated): Not set

.....000000000000000000000000 = Reserved bits: 0x00000000

Internet Protocol Version 4, Src: 192.1.1.1, Dst: 153.16.59.224

User Datagram Protocol, Src Port: 4342, Dst Port: 4342

Locator/ID Separation Protocol

0001 = Type: Map-Request (1)

.....000000.. = Flags: 0x00

0000 00 04 00 01 00 06 08 00 27 db d1 5c 00 00 08 00 '\.....

0010 45 00 00 5c 00 1e 40 00 ff 11 a3 fc c0 01 01 01 E.. \..@.....

0020 93 53 83 20 10 f6 10 f6 00 48 1c 28 80 00 00 00 .S.H.(....

0030 45 00 00 3c 00 1d 40 00 ff 11 e5 a0 c0 01 01 01 E.<..@.....

0040 99 10 3b e0 10 f6 10 f6 00 28 6a 07 10 00 00 01 .;.....(j.....

0050 2c 42 6e b7 3d 93 53 04 00 01 0a 00 02 0e 00 01 ,Bn.=.S.....

0060 c0 01 01 02 00 20 00 01 99 10 3b e0 ;.....

Locator/ID Separation Protocol (lisp), 4 bytes

Packets: 10 · Displayed: 8 (80.0%)

Profile: Default

Figura 17

6 Conclusions

La implementació de LISP-DDT per a la plataforma OpenOverlayRouter, a data d'escriptura d'aquesta memòria, funciona correctament i complint tots els requisits necessaris. Per tant, es pot afirmar que els objectius del projecte s'han complert totalment.

Pel que fa a l'experiència personal, contribuir en un projecte com OpenOverlayRouter ha estat una experiència positiva. Durant la realització del projecte, he adquirit nous coneixements teòrics i pràctics en programació i desenvolupament de software. Durant els meus estudis d'Enginyeria Informàtica, sota l'especialització d'Enginyeria del Software, no havia tingut l'oportunitat de formar part d'un projecte real. L'experiència i responsabilitat de que el meu treball necessitava ser correcte per tal de que els usuaris presents i futurs de OpenOverlayRouter puguin fer-ne ús satisfactòriament, i no només per a obtenir una bona nota en alguna assignatura, ha estat enriquidora a nivell personal, i l'èxit obtingut en aquest projecte, m'ha aportat major confiança a l'hora d'incorporar-me al món laboral com a enginyer informàtic.

Bibliografia

- 1: V. Fuller, D. Lewis, V. Ermagan, A. Jain, A. Smirnov, Locator/ID Separation Protocol Delegated Database Tree, 2017
- 2: Cisco Systems, Locator/ID Separation Protocol (LISP) Virtual Machine Mobility Solution White Paper, 2014
- 3: Cisco Systems, Cisco Locator/ID Separation Protocol and Overlay Transport Virtualization Data Center Infrastructure Solutions for Distributed Data Centers, 2011
- 4: V.M. Moreno, F.M. Maino, D.L. Lewis, M.S.Smith, S.S. Sinha, LISP Deployment Considerations in Data Center Networks, 2014
- 5: <http://www.snijders-it.nl/2012/07/what-is-lisp-ddt.html>, What is LISP DDT?, 2012
- 6: <https://openoverlayrouter.org/>, OpenOverlayRouter: Programmable overlays,
- 7: Loránd Jakab, Albert Cabellos-Aparicio, Florin Coras, Damien Saucez, Olivier Bonaventure, LISP-TREE: A DNS Hierarchy to Support the LISP Mapping System, 2010
- 8: A. Cabellos, D. Saucez, An Architectural Introduction to the Locator/ID Separation Protocol, 2015